Coding Problem Analyzer: Detailed Features

1. Problem Classification

- Categorizes problems based on difficulty levels (Easy, Medium, Hard).
- Categorizes problems by topics (e.g., Arrays, Graphs, Strings, etc.).
- Identifies sub-topics such as Knapsack Problems, Tree Traversals, etc.
- Supports input from online platforms and custom datasets.

2. Problem Similarity Suggestions

- Uses Natural Language Processing (NLP) to analyze problem statements.
- Identifies similar problems based on semantic similarity and tags.
- Recommends 'Next Best Problems' to practice for consistent learning.

3. Automated Progress Tracking

- Tracks user progress across topics and difficulty levels.
- Visualizes progress with charts, heatmaps, and tables.
- Generates daily, weekly, and monthly reports for self-assessment.

4. Problem Analysis

- Analyzes time and space complexity of user-submitted solutions.
- Detects edge cases that might have been overlooked.
- Provides insights into solution patterns (e.g., recursion vs. iteration).

5. Solution Quality Grader

- Grades solutions based on correctness and efficiency.
- Compares user solutions with optimal ones.

- Evaluates code readability and adherence to best practices.

6. Custom Problem Generator

- Generates new coding problems with custom constraints.
- Allows users to define input size, topics, and edge case scenarios.
- Creates large-scale test cases for performance testing.

7. Topic Mastery Prediction (Advanced)

- Uses machine learning models to predict topic proficiency.
- Analyzes trends in user performance for dynamic recommendations.
- Highlights weak areas and suggests specific improvement strategies.

8. Personalized Learning Path

- Creates a dynamic roadmap based on user strengths and weaknesses.
- Focuses on weaker areas first to ensure well-rounded learning.
- Adapts suggestions based on user progress and feedback.

9. Integration with Online Platforms

- Fetches problems directly from platforms like LeetCode, Codeforces, etc.
- Syncs user submissions and progress from these platforms.
- Supports API-based problem fetching for seamless integration.

10. Gamification and Motivation

- Adds badges and rewards for milestones (e.g., solving 100 problems).
- Includes leaderboards to foster healthy competition among peers.
- Introduces time-limited challenges for a gamified experience.

11. Visualization of Progress

- Plots bar charts and line graphs for topic mastery.
- Generates heatmaps to identify focus areas.
- Shows historical performance trends for motivation.

12. Solution Feedback Mechanism

- Provides detailed feedback on user solutions.
- Highlights specific areas for improvement in code logic and structure.
- Suggests alternate algorithms or methods for problem-solving.